

ACCESIBILIDAD WEB INTEGRADA EN APLICACIONES MÓVILES. VALIDACIÓN EN UN CONTEXTO DE APPS TURÍSTICAS

ACCESIBILIDAD WEB INTEGRADA EN APLICACIONES MÓVILES

AUTORES: Josue Maidana Fornazarich¹Cintia Galain García²Sonia I. Mariño³DIRECCIÓN PARA CORRESPONDENCIA: jotaeze95@gmail.com

Fecha de recepción: 02-02-2020

Fecha de aceptación: 10-04-2020

RESUMEN

Actualmente, en una sociedad enfocada en la información y la comunicación se incrementa la importancia de la accesibilidad en el uso de las aplicaciones móviles. Se expone una propuesta que integra la evaluación de la AW en el nivel AA según el estándar de la W3C en la metodología D3A diseñada para aplicaciones móviles. La propuesta se validó en una App diseñada para difundir el turismo regional. Se exponen los principales resultados de aplicar las pautas de accesibilidad WCAG 2.1 en sus cuatro principios. Las conclusiones destacan la importancia de incluir enfoques centrados en todos los usuarios.

PALABRAS CLAVE: Sistemas informáticos; Web; Accesibilidad; Normas; Ingeniería de sistemas; Software.

WEB ACCESSIBILITY INTEGRATED IN MOBILE APPLICATIONS. VALIDATION IN A CONTEXT OF TOURIST APPS

ABSTRACT

Currently, in a society focused on information and communication the importance of accessibility in the use of mobile applications increases. A proposal is presented that includes in an adaptation of the D3A methodology designed for mobile applications, the evaluation of the AW in the AA level according to the W3C standard. In order to validate the proposal, regional tourism was chosen as the application context, and the results obtained by applying the WCAG 2.1 accessibility guidelines are shown.

KEYWORDS: Informatics systems, World Wide Web (WWW), Accessibility, Norms, Systems engineering, Software

¹ Universidad Nacional del Nordeste. Argentina.

² Universidad Nacional del Nordeste. Argentina. E-mail: solgalain@gmail.com

³ Licenciada en Sistemas. Magíster en Informática y Computación. Magíster en Epistemología y Metodología de la Investigación Científica. Profesora Titular, Dedicación Exclusiva, del Departamento de Informática. FACENA, UNNE Antecedentes en la formación de Recursos Humanos de grado y posgrado. Dirección de proyectos de investigación aplicada e I+D. <https://orcid.org/0000-0003-3529-7003> Universidad Nacional del Nordeste. Argentina. E-mail: simarinio@yahoo.com

INTRODUCCIÓN

Las tecnologías de la información y comunicación TIC abarcan todos los ámbitos de la vida cotidiana desde la educación, el comercio, la salud y el turismo, entre otros. Sin embargo, esto no implica que todas las personas puedan utilizar las diferentes aplicaciones debido a que “se suelen encontrar barreras que impiden o dificultan el acceso a los usuarios con limitaciones funcionales, a la sociedad de la información” [1].

En esta sección, se tratará sobre la accesibilidad que se considera un derecho para las personas sin importar su capacidad de tener facilidad al utilizar la tecnología móvil o web, se continuará exponiendo las pautas WCAG 2.1 para su posterior aplicación en el desarrollo de una aplicación móvil orientada al turismo e indicando las herramientas software elegidas para corroborar la accesibilidad.

1.1 Accesibilidad y accesibilidad web

Se define la Accesibilidad al “conjunto de características que debe disponer un entorno, producto o servicio para ser utilizable en condiciones de confort, seguridad e igualdad por todas las personas y, en particular, por aquellas que tienen alguna discapacidad” [2]. La accesibilidad al medio físico y comunicacional es un derecho a ejercer por todos los ciudadanos independientemente de sus capacidades, por lo que este concepto se relaciona con el derecho de las personas [3].

Por lo expuesto se deben desarrollar aplicaciones móviles accesibles teniendo en cuenta las necesidades de todos los usuarios, buscando alcanzar un diseño universal que permita lograr la inclusión digital, mejorando las condiciones de uso y la calidad de vida de todos los usuarios. La accesibilidad web se refiere en general a hacer sitios web y aplicaciones más accesibles a las personas con discapacidad cuando se utilizan dispositivos móviles [4, 24, 25].

En la República Argentina rige la ley 26.378 “Convención sobre los derechos de las personas con discapacidad” aprobada por la Asamblea General de las Naciones Unidas, el cual dice en el artículo 9° del anexo I “Promover el acceso de las personas con discapacidad a los nuevos sistemas y tecnologías de la información y las comunicaciones, incluida Internet” [5]. Propuestas como la que se expone aportan al cumplimiento del artículo de la Ley.

La Iniciativa de Accesibilidad a la Web (WAI) del World Wide Web Consortium (W3C) estableció las pautas WCAG 2.1 [6, 27, 28] que se organizan en principios con pautas asociadas de acuerdo a un nivel de conformidad (A, AA, AAA). Sus principios son los siguientes Perceptible, Robusto, Operable y Comprensible [6].

1.2 Metodologías móviles

La Informática se compone de nueve disciplinas, siendo una de ellas la Ingeniería del Software (IS) [7]. Posee como elementos clave los métodos, las herramientas y los procedimientos que facilitan controlar el proceso de construcción del software y brindan a los desarrolladores las bases de la calidad de una forma productiva [8, 9].

La tecnología móvil permite mayor portabilidad e interacción con potenciales usuarios. El desarrollo de aplicaciones móviles difiere del ciclo de vida tradicional del software por lo que se requiere el estudio y elección de metodologías específicas para diseñarlas y construirlas que fortalezcan el funcionamiento de la aplicación. En este artículo se optó por una metodología con un ciclo incremental que facilitó el desarrollo de la aplicación propuesta.

DESARROLLO

Se adaptó la metodología D3A [10], que se compone de siete fases siendo los resultados preliminares expuestos en [26, 27, 29]. Notando la importancia de generar soluciones móviles accesibles, se adaptó la propuesta inicial y se realizaron tres incrementos, en cada uno de ellos se generó un prototipo funcional. La Fig. 1 muestra el ciclo de vida de la metodología D3A adaptado a esta propuesta.

El primer incremento abarcó la creación de la base de datos del servidor y las funcionalidades de registro de los usuarios visitantes, inicio de sesión de los usuarios registrados y el diseño de la interfaz del menú principal destinada a todos los perfiles de usuarios, descrita en [26].

En el segundo incremento se refinaron las interfaces a fin de asegurar su correcta utilización. Se definieron las listas de los sitios para cada opción del menú principal, y la interfaz de información útil. Se agregó la opción para indicar “me gusta” a cada sitio. El usuario registrado dispone de una interfaz para establecer sus datos y configuraciones de accesibilidad. Se crearon las funcionalidades para el usuario administrador, para agregar, eliminar y actualizar los sitios de interés. Se introdujo la validación de AW

En el tercer incremento se mejoraron las funcionalidades incluidas en el incremento anterior. El usuario registrado puede solicitar agregar nuevos sitios, se crearon las interfaces del detalle de cada uno de los sitios (actividades, atracciones, compras, hospedajes y gastronomía) que permiten agregar comentarios, ver la galería de fotos del sitio, guardar y visualizar los sitios como favoritos, agregar fotos a la galería, visualizar los horarios y dirección de cada sitio y se implementó la funcionalidad de localización GPS para ubicar los diferentes lugares. El usuario administrador puede visualizar los informes.

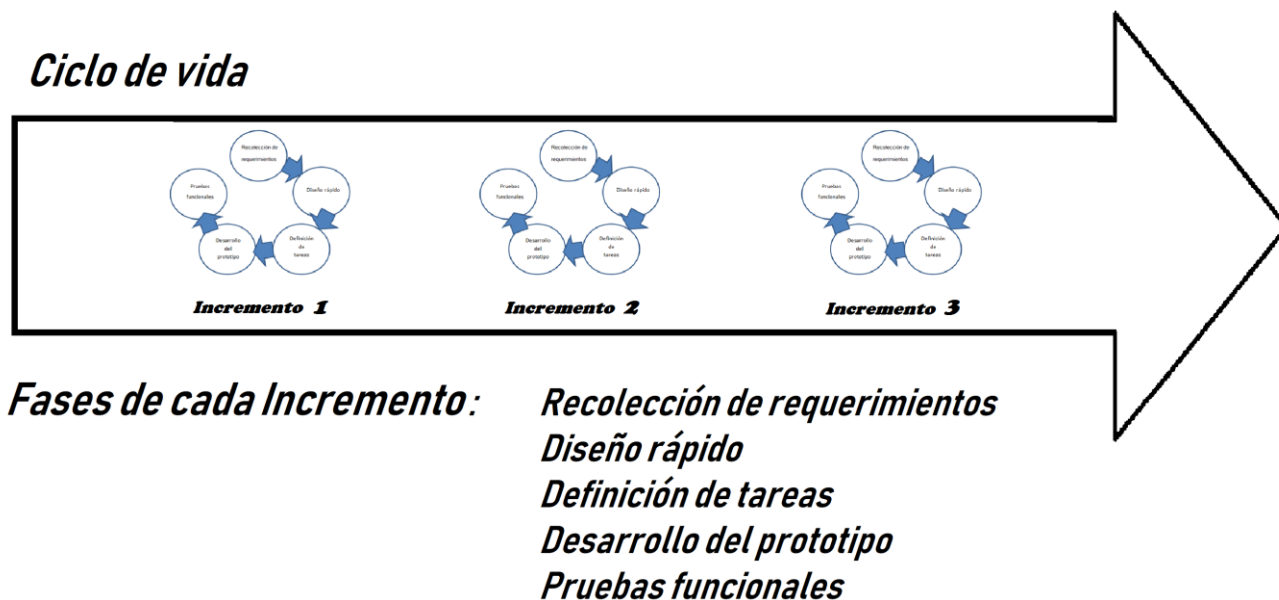


Fig. 1 Ciclo de vida metodología D3A. Fuente: elaboración propia adaptado de [10].

Primera fase: Recolección de requerimientos

Se establecieron los requerimientos funcionales y no funcionales de la aplicación. En [23] se propone contemplar a la AW como un requerimiento no funcional que debe estar presente en todos los desarrollos software

Segunda fase: Diseño rápido

Se definió un primer prototipo funcional destinado a los desarrolladores del equipo de trabajo. Al comienzo los prototipos se realizaron a mano, en papel y después se construyeron mediante la herramienta NinjaMock [11].

Tercera fase: Definición de tareas

En esta fase se aplicó la metodología Kanban [12] para organizar las tareas en un tablero y visualizar sus estados. El flujo del proyecto se basó en el principio del desarrollo incremental, por lo que sus actividades son divididas en distintas partes para agilizar el proceso. Se realizaron tarjetas visuales, cada tarea se escribió en una tarjeta y se indicó en el tablero la fase a la que corresponde. Para implementar el tablero Kanban, se utilizó la herramienta Trello [13].

Cuarta fase: Desarrollo del Prototipo

Se desarrolló el prototipo y su codificación utilizando la plataforma Android Studio [14], [15], [16], y la base de datos Firebase [17]. Se aplicaron pautas descriptas en el apartado 1.1.1 de accesibilidad en el desarrollo móvil.

Quinta fase: Pruebas funcionales

Se realizaron las pruebas para los incrementos. Se elaboró un plan de prueba por cada caso de uso y se abordó análisis de accesibilidad a nivel AA según el estándar W3C. Se utilizaron las herramientas Lint [18] y la aplicación de Google “Test de Accesibilidad para Android” [19].

Estas se aplicaron y evaluaron desde el comienzo del desarrollo, se describe el análisis en el siguiente apartado incluyendo las interfaces de la aplicación final.

3. Resultados

Se exponen los principales resultados de integrar en la metodología elaborada por Technosite [20, 21] y adaptada en [26] las Pautas de Accesibilidad al Contenido en la Web del W3C/WAI [6, 25]. Se estableció cumplimentar una adaptación del nivel AA de accesibilidad para el desarrollo móvil en Android considerando los principios de perceptibilidad, operatividad, compresibilidad, robustez. Se aplicó el test de accesibilidad en distintas las primeras etapas del desarrollo. A continuación, se enuncian los principios y las pautas elegidas en este trabajo:

Primer principio: Perceptible. La información y los componentes de la interfaz del usuario deben presentarse a los usuarios de forma que puedan percibirlos. En el desarrollo se consideraron:

- Características sensoriales: Las instrucciones proporcionadas para comprender y operar el contenido no dependen únicamente de la forma, el color, el tamaño, la ubicación visual, la orientación o el sonido.
- Propósito de los controles: Colocar marcadores con nombres convencionales de los campos de formulario, botones convencionales o controles, o enlaces.

- Orientación: El contenido no restringe su vista y funcionamiento a una única orientación de visualización, como vertical u horizontal, a menos que una orientación de visualización específica sea esencial.
- Identificar propósito de entrada: Indicar el propósito de cada campo de entrada que recopila información sobre el usuario.
- Distingible: Facilita a los usuarios ver contenido, incluyendo la separación de los primeros planos del fondo.
- Uso del color: El color es uno de los medios visuales utilizados para transmitir información.
- Contraste mínimo: La presentación visual de texto e imágenes de texto tiene una relación de contraste de al menos 4-5:1, excepto en textos grandes (deben ser 3:1), textos decorativos y logotipos.
- Imágenes de texto: Se utiliza texto para transmitir la información en vez de imágenes de texto, excepto en textos esenciales o imágenes de texto configurables.
- Contraste no textual: La presentación visual (imágenes, botones, hipervínculos, etc.) tiene una relación de contraste de al menos 3: 1 frente a color/es adyacente/s.

Segundo principio: Operable. Los componentes de la interfaz de usuario y la navegación deben ser operables. Se aplicaron las pautas relacionadas con:

- Convulsiones y reacciones físicas: No diseñar el contenido de manera que provoque convulsiones.
- Tres flashes o debajo del umbral: No contener nada que parpadee más de tres veces en un periodo de un segundo.
- Navegable
 - Bloques de derivación: Permitir a las personas que navegan secuencialmente a través del contenido acceder directamente a la información principal.
 - Página titulada: Las páginas tienen títulos que describen el tema o el propósito.
 - Orden de enfoque: Se debe poder recorrer secuencialmente el contenido por ejemplo en un formulario.
 - Foco visible: Indicar visualmente donde está el foco de escritura.

Tercer principio: Comprensible. La información y el funcionamiento de la interfaz del usuario deben ser comprensibles. Se contemplaron las pautas:

- Legible: El contenido de texto debe ser legible y comprensible.
- Predecible: La aplicación debe operar de manera predecible al recibir entradas, es decir, el cambio de estado en la interfaz de usuario no provoca automáticamente un cambio en el contexto a menos que el usuario haya sido advertido de ese comportamiento.

Cuarto principio: Robusto. El contenido debe ser lo suficientemente sólido como para que pueda ser interpretado por una amplia variedad de agentes de usuario, incluidas las tecnologías de asistencia. Se consideraron las siguientes pautas:

Compatible. Maximice la compatibilidad con los agentes de usuario actuales y futuros, incluidas las tecnologías de asistencia.

Atendiendo a lo expuesto precedentemente, se sintetizan los principales resultados de aplicar las pautas seleccionadas de cada uno de los principios en la aplicación móvil.

Primer principio: Perceptible

El diseño de la aplicación se realizó asegurando que cada texto del botón de la aplicación disponga de información de la función que realiza y no dependa de un color o imagen representativa, de forma que el usuario identifique sencillamente el propósito de los botones, como se puede apreciar en la Fig. 2.



Fig. 2 Ejemplo del diseño de los botones

La orientación de cada interfaz se diseñó para cumplir su función de la manera más apropiada. En las interfaces de inicio de sesión y de registro, se optó por una orientación vertical para una mejor adaptación del formulario y el usuario ingrese sus datos de forma más cómoda.

Cada elemento editable para recopilar información, como inicio de sesión, registro o modificar datos del usuario, entre otros, tiene vinculado una etiqueta descriptiva que especifica la entrada esperada. Esto se lleva a cabo usando el método de Android Studio llamado hint, como se puede visualizar en la Fig. 3. Con la herramienta Lint, se corroboró automáticamente la existencia de un campo editable en el código de la aplicación que no contenga el método (Fig. 4).

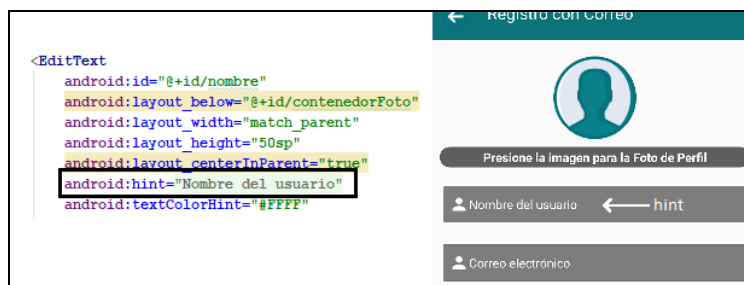


Fig. 3 Ejemplo de la aplicación del atributo hint en el código y su resultado

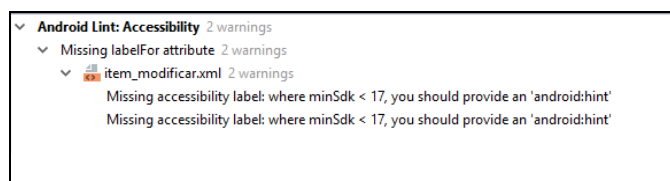


Fig. 4 Ejemplo de la ejecución de Lint mostrando atributos hint faltantes

Los colores elegidos en la aplicación aseguran un buen contraste de 4-5:1 o 3:1 (para textos o elementos visuales respectivamente, según el nivel AA) entre los colores de los elementos principales y el color del fondo. Se comprobó aplicando el análisis interfaz por interfaz con la aplicación Test de Accesibilidad de Android, un ejemplo de las pruebas se puede corroborar en la Fig. 5.

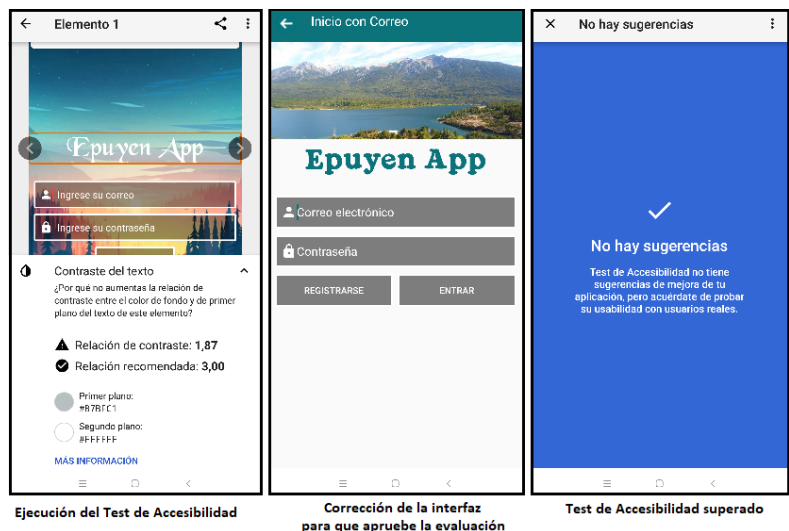


Fig. 5 Ejemplo de ejecución, corrección y aprobación del test

Segundo principio: Operable

Para que la aplicación no provoque convulsiones y reacciones físicas, debe carecer de algún elemento visual que parpadee durante su uso. Personas con discapacidades fotosensibles pueden sufrir ataques por cambios rápidos de luz a oscuridad, que superen cierta frecuencia y tamaño.

Cada interfaz en la aplicación posee un título de manera que represente el tema o el propósito de la información contenida. Además, esto ayuda a los usuarios que recorren la aplicación saber la interfaz en que se ubican. Se agregaron botones en la parte superior para que las personas regresen al menú principal, una vez accedido de forma secuencial hasta la información de un sitio facilitando el uso y acceso a la información de la aplicación.

Cuando los campos editables reciben el foco o gesto del usuario, se indica visualmente en qué campo el usuario está por escribir, para facilitar su interacción con la aplicación. Estas características se pueden observar en la Fig. 6.

Las interfaces con formularios aseguran que el recorrido de cada campo de entrada de texto es secuencial y ordenado. En Android Studio se definió en cada campo editable el atributo `inputType`, de modo que en el teclado aparezca el símbolo para que derive al siguiente campo editable (Fig. 7)

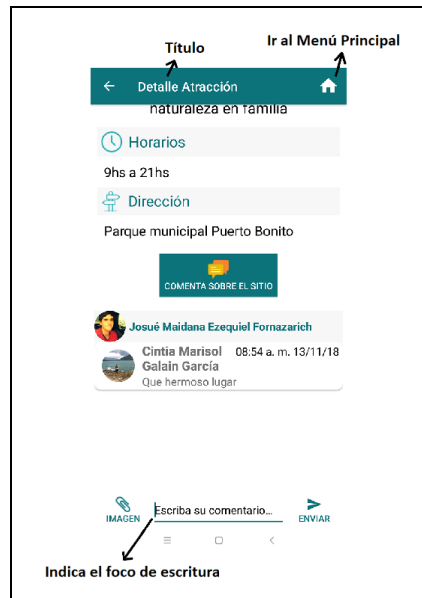


Fig. 6 Ejemplo de una página titulada, del foco visible y bloques de derivación

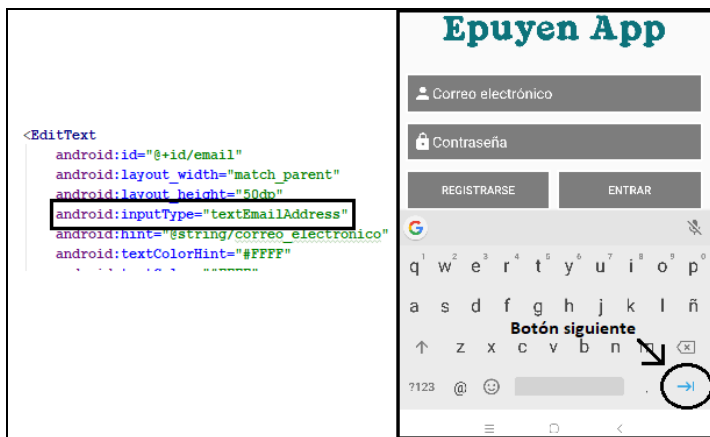


Fig. 7 Aplicación del inputType y su resultado

Tercer principio: Comprensible

Para que la aplicación sea comprensible y fácil de usar, se buscaron diseños simples, con la información justa para evitar saturar al usuario, sin agregar complejidad en el recorrido de la aplicación y especificando la ubicación actual. La Fig. 8 ilustra un diseño simple de la aplicación.



Fig. 8 Ejemplo del diseño simple de la aplicación

La aplicación debe ser predecible, por lo tanto, cada botón tiene la descripción de la función que lleva a cabo. Cuando el usuario presiona o activa el botón, la aplicación despliega un mensaje en la interfaz sobre la acción en ejecución. En Android Studio esta función se implementa con la clase Toast, muestra mensajes al usuario sin interferir con las acciones que realiza. La creación de estos mensajes se ejemplifica en la Fig. 9.

```
Toast.makeText(Bienvenida.this, "Bienvenido/a "+nombre, Toast.LENGTH_SHORT).show();
```

Fig. 9 Ejemplo de la utilización del Toast

Cuarto principio: Robusto

Cualquier contenido debe ser interpretado por una amplia variedad de agentes de usuarios o tecnologías de asistencia. Para que la aplicación sea compatible con esas tecnologías se aplicó el método setContentDescription de Android Studio si la ejecución es dinámica o el método contentDescription en caso estático (Fig. 10). Estas etiquetas permiten a los lectores de pantalla, proporcionar una explicación particular a los usuarios que dependen de estos servicios. Además, con la herramienta Lint, se corroboró de forma automática la existencia de un elemento visual (botón, imagen, campo de escritura, etc.) en todo el código de la aplicación que carezca del método, como se ilustra en la Fig. 11.



Fig. 10 Ejemplo de uso de la descripción del contenido

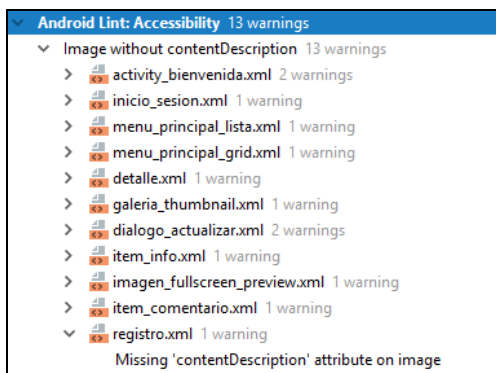


Fig. 11 Ejemplo de la ejecución de Lint mostrando atributos contentDescription faltantes

CONCLUSIONES

A nivel mundial existe una marcada tendencia en el diseño y construcción de aplicaciones móviles y además la Accesibilidad desde las TIC es una cuestión emergente. En este trabajo se diseñó y validó un procedimiento que basado en la metodología D3A incorpora cuestiones que aseguran la accesibilidad web según el estándar W3C nivel AA. Lo expuesto asegura que la aplicación móvil pueda utilizar todos los sujetos independientemente de sus capacidades mediante las características descriptas.

En referencia particular a la aplicación descrita para validar la solución se afirma que los avances de la tecnología móvil permiten ofrecer diferentes soluciones en vastos dominios de conocimientos, siendo una de ellas, el turismo. La App descrita puede adecuarse a otros destinos turísticos con grandes atractivos naturales para facilitar su difusión.

REFERENCIAS BIBLIOGRÁFICAS

Alonso F. Libro Blanco: Por un nuevo paradigma, el Diseño para Todos, hacia la plena igualdad de oportunidades. Extraído de: https://ddd.uab.cat/pub/estudis/2007/hdl_2072_4720/LIBRO_BLANCO_AcCESIBILIDAD_2003.pdf

Amovil. Libro blanco para el diseño de Tecnología Móvil accesible y fácil de usar. Extraído de: <http://www.amovil.es/sites/default/files/e-51libroblancoespanol.pdf>

Android, “Android Developers, Tools: Android Studio”. Extraído de: <http://developer.android.com/sdk/index.html>.

Android. Android Developers Dashboards. Extraído de: https://developer.android.com/about/dashboards/index.html?utm_source=suzunsuz.

Firebase. Extraído de: <https://firebase.google.com>.

Galain Garcia, C., Maidana, J. y Mariño S. I. 2018, Epuyen turística. Una APP para los turistas 2.0, Revista TURYDES. 11(24). Extraído de: <https://www.eumed.net/rev/turydes/24/epuyen-turistica-app.html>

Ley 26.378. Convención sobre los derechos de las personas con discapacidad. Extraído de: <http://servicios.infoleg.gob.ar/infolegInternet/anexos/140000-144999/141317/norma.ht>.

Lint. Extraído de: <https://developer.android.com/studio/write/lint?hl=es-419>.

López J., Moreira J., Alava N., Metodología para valorar y clasificar herramientas de evaluación de accesibilidad. Extraído de: <https://www.redalyc.org/jatsRepo/4768/476854589010/html/index.html>.

Maidana J., Galain Garcia C., Mariño S. I. (2017). Metodología D3A para soluciones móviles. Un caso de estudio aplicado a la promoción turística. Revista Publicando. 4(1), 21-39.

Mariño S. I., Alfonzo, P. L. (2018). An empirical WCAG 2.0 guidelines evaluation applied to journal management software. International Journal of Recent Engineering Research and Development. 3(11). 16-22.

Mariño S., Alfonzo P., Galain C., Maidana J., Alderete R., Accesibilidad Web, aportando a la inclusión. (2018). Trabajo presentado en el XX Workshop de Investigadores en Ciencias de la Computación, Corrientes, Corrientes.

Mariño S., Godoy M., Alfonzo P., Acevedo J., Gómez S., Fernández L., Vázquez A. (2012). Accesibilidad en la definición de requerimientos no funcionales. Multiciencias. 12(3), 305-312

Mariño S.I., Alfonzo P.L., Galain, C., Maidana, J., Alderete, R. Y. (2018). Accesibilidad Web, Aportando a la inclusión. Trabajo presentado en el XX Workshop de Investigadores en Ciencias de la Computación, Corrientes, Corrientes.

Metodología de Technosite. La metodología del observatorio de Infoaccesibilidad. Extraído de: <https://www.discapnet.es/areas-tematicas/tecnologia-inclusiva/observatorio-de-accesibilidad-tic/metodologia-del-observatorio>

Mobile Accessibility. Extraído de: <https://www.w3.org/WAI/mobile/Overview.html>.

NinjaMock. Extraído de: <https://ninjamock.com/>.

PhoneGap. Extraído de: <http://www.phonegapspain.com>

Ponce U., Yañez V., Soto R. (2014). Propuesta metodológica para el desarrollo de aplicaciones móviles para dispositivos Android. Trabajo presentado en el Congreso Internacional de Investigación, Tabasco, México.

Pressman R. (2010). Ingeniería del Software. Un enfoque práctico. Editorial Mc Graw Hill.

Sommerville I. (2011). Ingeniería del Software. Editorial Pearson Educación.

Tablero Kanban. Extraído de: <http://kanbantool.com/es/tablero-kanban>.

Technosite. Extraído de: <https://tecnoaccesible.net/node/118>.

Test de accesibilidad. Test de Accesibilidad de Google para Android. Extraído de: <https://play.google.com/store/apps/details?id=com.google.android.apps.accessibility.auditor&hl=es>.

Thomas G. P. (2013). Ingeniería de Software en el desarrollo de Aplicaciones para Dispositivos. Trabajo presentado en el XV Workshop de Investigadores en Ciencias de la Computación, Paraná, Entre Ríos.

Toledo G. (2013). Accesibilidad digital para usuarios con limitaciones visuales. Extraído de: http://sedici.unlp.edu.ar/bitstream/handle/10915/24871/Documento_completo__%20Toledo%20Gabriela%20UNLP11-2012modif.pdf?sequence=1

Trello. Extraído de: <https://trello.com>.

W3C. Designing for Inclusion. Extraído de: <https://www.w3.org/WAI/users/>.

W3C. Web Content Accessibility Guidelines. Extraído de: <https://www.w3.org/TR/WCAG21/>.

